A METHOD AND SYSTEM FOR GENERATING AND VERIFYING A KEY PROTECTION CERTIFICATE

Field of Invention

The present invention relates to a data processing system and method for generating a unique digital certificate within the secure domain of a personal security device (PSD). The generated certificate is used by another party to verify that cryptographic keys are bound to and protected by a specific PSD.

Background of Invention

The current art involving personal security devices (PSD) including smart cards, subscriber identification modules (SIM), wireless identification modules (WIM), identification tokens and related devices are designed to afford enhanced protection of asymmetric private keys and shared secret symmetric keys over that provided by software solutions or other means.

PSDs also provide improved support of digital signature applications for non-repudiation purposes than is afforded using software solutions or other means. For non-repudiation purposes, it is essential that private and secret keys be protected by the security mechanisms implemented within a PSD and not be disclosed. This is a basic foundational premise behind the various public key infrastructures available in the current art and as such is an area vulnerable to compromise by a sophisticated attacker as described below.

Currently, there are two methods in which cryptographic keys are installed within the secure domains of a PSD. The first method involves placement of cryptographic seed information inside a PSD which is then used to generate cryptographic keys based upon internal key generating algorithms. A second method involves directly injecting the cryptographic keys into a secure domain of the PSD and storing the keys in accordance with the security policies included in the PSD.

Neither method generates any proof to another party that the cryptography keys are securely stored and bound to the PSD in which they were intended. The keys exist essentially independent of the device in which they are stored. Users and third parties therefore implicitly rely upon the security of the installation process to ensure that the

cryptographic keys have been installed in the correct PSD and have not been replaced or duplicated in another unauthorized device.

Until recently, the generation of cryptography keys in PSDs occurred using end-to-end security mechanisms. The card issuer controlled all aspects of key generation and card issuance, which provided little opportunity for unauthorized disclosure of private or secret keys. However, as demand for increased security mechanisms and storage of multiple credentials on a single card has developed, the use of end-to-end security mechanisms is being replaced with remote post issuance methods.

For example, US Patent 6,005,942 describes a post issuance method of installing applications into a smart card. This method allows an authorized entity other than the original card issuer to install applications including proprietary information such as cryptographic seeds, private keys and symmetric keys into a secure domain of the card.

The implicit assumption utilizing this method is that the existing cryptographic keys employed during the post issuance installation process have not been compromised. Thus, it becomes possible for an authorized provider, unaware that the original cryptographic keys have been compromised, to operatively install additional proprietary information into an unauthorized card. There currently is no method for verifying that a particular key is bound to a particular device.

Summary of Invention

This invention provides a method for generating a unique key protection certificate, which provides assurances to another party that private or secret (symmetric) keys are securely stored within the PSD. The certificate is generated using the cryptographic services and data processing capabilities normally provided with the current generation of PSDs.

A typical PSD, for example a smart card, contains a microprocessor for executing programmatic instructions, read only memory (ROM) for containing essential programs such as a runtime environment and security policies, non-volatile memory for storage of information using electrically erasable programmable read-only memory (EEPROM) and lastly volatile random access memory (RAM) for temporary storage of information.

The installed security policies and added security extensions generally support standardized cryptographic methods including asymmetric key methods such as DSA, RSA, or both, symmetric key methods such as DES, 3DES or both, non-keyed message digest methods such as MD5, SHA-I or both and keyed message digest methods such as MAC.

PSDs are also configurable to allow separate secure domains allowing different providers to store proprietary information including symmetric and asymmetric keys. Each domain allows access to common utilities and services installed in the PSD but the PSD's security policies prevent accessing of secure information installed outside of a providers allocated domain. Thus, multiple sets of separately accessible keys may exist within a PSD at any one time but only the owner of the keys may access the domain in which they are installed.

By using the established security policies and added extensions employed in a typical PSD, it is possible to generate a key protection certificate, which provides greater assurances to another party that private or secret keys are bound to and protected by a PSD.

To practice this invention, a key protection certificate generating algorithm is loaded into a common domain of a PSD and operatively stored in either the system ROM during masking or in non-volatile RAM. This algorithm operates sequentially with installed cryptographic key generating algorithms and a pre-encoded device name (usually the PSD's serial number) to produce a unique digital certificate upon completion of cryptographic key generation. The digital certificate is then available for verification.

In the preferred embodiment of the invention, an additional set of parameters is generated which includes contextual attributes related to the PSD and a particular private or symmetric key generated within the PSD. The contextual attributes may include but are not limited to a counter, trusted time source, the date and time of generation, version numbers, applications the key is intended to be used for, key life cycle information (expiration date, etc.), cryptography methods employed, key identification labels and receiving party identification information.

A portion of the contextual attributes are intended to be available as clear text in the key protection certificate along with the device name for review and verification by a receiving party. The remaining portion of the contextual attributes are obfuscated using a

second symmetric encryption method and key. Some or all of these attributes may also be used to reference or diversify the shared secret keys by the receiving party. The receiving party may be a second party who is seeking direct verification of the sending party's credentials using for example pretty good privacy (PGP) techniques or a trusted third party certificate authority (CA) who provides the verification as part of an organized service using for example X.509 formatted certificates. For simplicity, "receiving party" will be used to refer to both a second party and trusted third party.

In the preferred embodiment of the invention, the key protection certificate is produced by encrypting a portion of the contextual attributes with a first shared secret key, signing a device name (or derivation thereof) with a private key and concatenating the encrypted portion of the contextual attributes, clear text portion of the contextual attributes, clear text device name and signed device name producing an intermediate result. This intermediate result will be used by a receiving party to verify that the private key is securely stored in the PSD. Additionally, the intermediate result may be used to prove that the result was generated within the secure domain of the PSD.

A message authentication code (MAC) function is then performed on the intermediate result and encrypted using a second shared secret key. The second shared secret key is a symmetric key known only to the PSD and the receiving party. The results of the MAC are then concatenated with the intermediate result described above, producing the key protection certificate. The MAC portion will be used by a receiving party to verify that the device private key was generated within the secure domain of the PSD. Once produced, the certificate is available for verification by a receiving party.

A receiving party verifies the certificate by cross referencing the device name (or derivation thereof) with the proper public contextual information, secret keys, public key, cryptographic algorithms, reference parameters, etc. contained in a database, lookup table or similar arrangement. Once the proper access information is determined, the verification is performed by decrypting the device name portion of the certificate using the complementary public key and comparing the result to the plain text version of the device name This operation confirms to the receiving party that the transaction occurred using the proper key pair.

Next, the receiving party, using the same MAC algorithm and shared secret key generates a duplicate MAC. The generated MAC is then compared to the MAC contained in the certificate. An exact match between the independently created MAC and the

received MAC provides assurances that the transaction occurred within the secure domain of the PSD.

Lastly, the private contextual attributes are decrypted using a second shared secret key and compared to reference parameters securely shared between the PSD and the receiving party. An exact match of these parameters provides further assurances that the certificate was validly generated. The parameters maintained by the receiving party may be a counter which increments each time the certificate is verified, a trusted time stamp or another variable controlled by the receiving party which is securely shared with the PSD.

New digital certificates may be generated each time an authorized change is made to any of the cryptography algorithms or keys contained within the PSD. In the preferred embodiment, context attributes are updated and stored as part of the digital certificates.

It should be understood to those familiar with the art that more than one digital certificate and associated cryptography keys might be stored within the secure domain of a PSD for servicing different receiving parties. The number of digital certificates and associated cryptography keys is limited only by available memory resources. In the preferred embodiment of the invention, the key protection certificate is intended to conform to the X.509 and/or ANSI X.9 certificate format standards for use by a trusted third party certificate authority.

Furthermore, different combinations of asymmetric and symmetric keys, signed and unsigned message digest functions, and other information may be employed to generate and validate an equivalent key protection certificate. For example, a null vector could be signed with the private key rather than using the device name. Other combinations involving the use of shared secret keys and a private key will work as well.

Brief Description of Drawings

FIG. 1 - is a general system block diagram for implementing present invention.

FIG. 2 - is a detailed block diagram illustrating the digital certificate generating process.

FIG. 3A - is a detailed block diagram illustrating the first part of the validation process.

FIG. 3B - is a detailed block diagram illustrating the second part of the validation process.

FIG. 3C - is a detailed block diagram illustrating the third part of the validation process.

FIG. 4 - is a detailed block diagram illustrating final part of the validation process.

Detailed Description of Preferred Embodiment

In this invention, a key protection certificate is created and stored sequentially in conjunction with cryptographic key generation. In the preferred embodiment of the invention, a shared secret key is securely injected into a PSD during or after personalization. In one embodiment of the invention, the secret key is shared with a second party who will perform the direct verification of the digital certificate when received. In another embodiment of the invention, the secret key is shared with a trusted third party certificate authority who performs the verification of the certificate and informs a third party of the validly of the certificate in the form of an X.509 and/or ANSI X.9 formatted certificate.

Referring to FIG. 1, a typical arrangement of a PSD 40 is depicted where separate domains (Domain 1 45, Domain 2 50, Domain 3 55 through Domain n 60) are established allowing unrelated service providers to install and maintain provider specific sets of asymmetric public 15 and private keys 10, a first shared secret key (MAC) 5 used during the encryption of the message digest, a second shared secret key (Encrypt) 95 used for encrypting the private portion of the contextual attributes, and key protection certificate 20.

A unique device name 65 is generated during the PSD manufacturing process, which is common and accessible to all domains but unalterable for the life of the PSD. Cryptographic algorithms 70, including symmetric 25 and asymmetric 30 key generating modules, a message authentication code module 85, the added key protection certificate

module 90 and asymmetric decryption module are contained in an API layer and are likewise common and accessible to all domains. These modules are used for generating the cryptographic information stored in each of the providers secure domain. Another layer 75 contains cryptographic seed information for generation of cryptographic keys.

PSDs follow a layered structure in which an applications programming interface (API) rides above a runtime-operating environment 80. In the preferred embodiment of the invention, the layers below the API layer are unmodified and thus not included in the basic depiction.

In FIG. 2, a detailed block diagram of the digital certificate generating process is depicted. To generate the key protection certificate 20, an initial set of contextual attributes is generated 270AB. A portion of the initial contextual attributes are then encrypted with a first shared secret key 95 forming a private and public set of contextual attributes, followed by signing the device name 65 with a private key 10. The device name may be the PSD serial number or name derived from the serial number. The signed device name 210 is then concatenated 220 with the clear text device name 65, public contextual attributes 270B and private contextual attributes 270A to produce a first intermediate result 230.

The first intermediate result 230 is then processed using a message authentication code 240 and a second shared secret symmetric key 5 producing a second intermediate result 245. The second intermediate result 245 is then concatenated 250 with the first intermediate result 230 producing the key protection certificate 20. Once generated, the certificate is available for validation by a receiving party.

In FIG. 3A, a detailed block diagram of the first part of the multi-step process employed to validate the key protection certificate 20 is depicted. Upon receiving the key protection certificate 20, the portions of the digital certificate containing the plain text device name 65 and signed device name 210 are extracted. The device name 65 is used to cross-reference the required cryptographic keys, algorithms and reference parameters necessary to perform the validation process.

Once the proper access information has been determined, the signed device name 210 is decrypted 305 using the complementary public key 15 resulting in an unverified device name 65'. The unverified device name 65' is compared 315 to the extracted device name 65. If the results are equal 320, then the private key has been

validated. Otherwise 310, the private key has somehow been altered and a failure flag is set in the certificate identifying the invalid key pair validation step.

Referring to FIG. 3B, the next part of the validation process generates an independent message authentication code (MAC) using a method authentication code 325 identical to that 85 implemented in the PSD. The MAC uses the portions of the certificate containing the private contextual attributes 270A, public contextual attributes 270B, device name 65, signed device name 210 and the second shared secret key 5. The resulting message authentication code 340 is compared 345 with the message authentication code 260 contained in the certificate 20. If the results are equal 355, then the key generation process can be assumed to have occurred within the secure domain of the PSD (since only the PSD and the receiving party should possess the shared secret key,) and this portion of the key generating process is validated. Otherwise 350, the key generating process may not have occurred within the secure domain of the PSD and a failure flag is set in the key protection certificate identifying the invalid key generation location step

Referring to FIG. 3C, the third part of the validation process decrypts 360 the private contextual attributes 270A using the first shared secret key 95 resulting in a clear text version of the private contextual attributes 270A'. One or more parameters included in the private contextual attributes 270A' is compared 370 against reference parameters 375 maintained by the receiving party. If the results are equal 380, then the contextual attributes have been validated. Otherwise 365, the key protection certificate may not be valid and a failure flag is set in the key protection certificate identifying the invalid contextual attribute step.

Referring to FIG.4, the results of the preceding validation processes are summarized. If a valid device name has been verified 410 as described in FIG.3A 320, then it is verified that a valid MAC has been obtained as described in FIG. 3B 355. If a valid MAC has been verified 420, then it is verified that valid contextual attributes have been verified as described in FIG. 3C 380. If valid contextual attributes have been verified 430, then the key protection certificate 435 is fully validated and should be accepted by the receiving party. If validation is being performed by a trusted third party certificate authority, no failure flags should be set and the validated digital certificate should be forwarded to the receiving party for acceptance.

If any of the three validation steps fail, then the key protection certificate should be rejected 440 by the receiving party. If validation is being performed by a trusted third party certificate authority, appropriate failure flags should be set and the failed digital certificate should be forwarded to the receiving party for rejection.

The foregoing described embodiments of the invention are provided as illustrations and descriptions. They are not intended to limit the invention to precise form described. In particular, it is contemplated that functional implementation of the invention described herein may be implemented equivalently in hardware, software, firmware, and/or other available functional components or building blocks. Other variations and embodiments are possible in light of above teachings, and it is not intended that this Detailed Description limit the scope of invention, but rather by the Claims following herein.